

高等职业教育

计算机类专业 规划教材

C#项目教程

(基础篇)

姚嵩 主编
王唯 侯俊芳 副主编



中国电力出版社
<http://jc.cepp.com.cn>

内 容 提 要

本书为高等职业教育计算机类专业规划教材。本书突破传统的教学模式，引入最新项目式教学理念，采用项目驱动式教学方法。本书结合 C#语言的特点，从基础的语法开始讲起，通过一个个项目逐步引导读者完成学习。《C#项目教程》分为基础篇和提高篇。本书为基础篇，介绍了 C#语言的语法、流程控制和数组、枚举等，重点强调 C#面向对象的编程思想，突出 C#的优点，如属性、索引、泛型、委托、事件等，并介绍了 .NET 的几个主要应用技术，如 Windows 窗体程序、ADO.NET 数据库应用程序和 ASP.NET Web 应用程序。

本书可作为高等职业院校相关专业教材使用，也可作为成人函授或培训班教材，还可供对 C#语言感兴趣的初学者学习和参考。

图书在版编目 (CIP) 数据

C#项目教程 (基础篇) / 姚嵩主编. —北京: 中国电力出版社, 2010.8

高等职业教育计算机类专业规划教材

ISBN 978-7-5123-0536-6

I. ①C… II. ①姚… III. ①C#语言—程序设计—高等学校: 技术学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 110874 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://jc.cepp.com.cn>)

航远印刷有限公司印刷

各地新华书店经售

*

2010 年 8 月第一版 2010 年 8 月北京第一次印刷

787 毫米×1092 毫米 16 开本 14.5 印张 349 千字

印数 0001—3000 册 定价 23.50 元

敬告读者

本书封面贴有防伪标签，加热后中心图案消失
本书如有印装质量问题，我社发行部负责退换

版权专有 翻印必究

前言

自.NET Framework 发布以来,业界掀起了学习.NET 技术的狂潮。C#和.NET 平台得以迅速普及,成为很多公司使用的主要技术之一。越来越多的应用程序都使用.NET 平台上的主流开发语言——C#进行开发。本书介绍了 C#语言的语法、流程控制和数组、枚举等,重点强调 C#面向对象的编程思想,突出 C#的优点,如属性、索引、泛型、委托、事件等,并介绍了C#的几个主要应用技术,如 Windows 窗体程序、ADO.NET 数据库应用程序和 ASP.NET Web 应用程序。

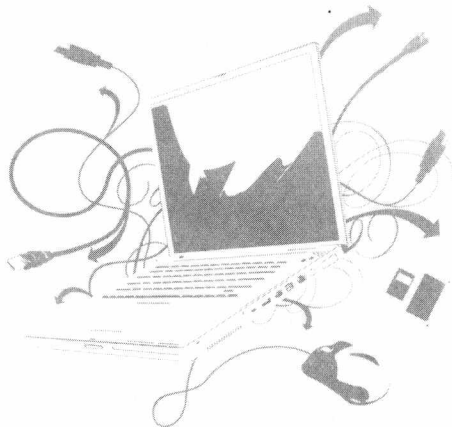
本书突破传统的教学模式,引入最新项目式教学理念,采用项目驱动式教学方法,结合C#语言的特点,从基础的语法开始讲起,通过一个个项目逐步引导读者完成学习。

为了保证教材的质量,我们邀请了几所高职院校在教学一线多年的老师共同编写此书,以便集思广益,尽量使内容丰富且准确无误。姚嵩老师编写了第1、2、3章及9、10章中的部分内容,王唯老师编写了第6、7章,张润华老师编写了第8章,侯俊芳老师、刘砚老师、武博文老师共同编写了第4、5、9、10章并完成了大量的文字处理以及代码调试工作。本书由薛宾、台雯主审。本书的部分示例借鉴了 Internet 上的公共文档中的内容,在此对那些无私提供文档的作者表示衷心地感谢。

限于作者的水平,加之时间仓促,书中难免有疏漏和不足之处,恳请专家、同仁和广大读者批评指正,在此先致以谢意。

编者

2010年5月



目 录

前言

第 1 章 C#语言概述	1
1.1 C#简介	1
1.2 .NET 应用程序框架	6
1.3 项目显示程序	9
第 2 章 C#语言基础	27
2.1 C#程序的结构	27
2.2 项目 1 判断字符串	30
2.3 项目 2 变量的使用	33
2.4 项目 3 计算 sinc 函数值	37
2.5 项目 4 流程控制	40
2.6 项目 5 预处理	46
第 3 章 调试与异常处理	48
3.1 项目 1 C#程序调试	48
3.2 项目 2 C#程序异常处理	53
第 4 章 面向对象程序设计	58
4.1 面向对象程序设计的基本概念	58
4.2 项目 1 类的定义与使用	59
4.3 项目 2 调用类的方法	64
4.4 项目 3 实现继承与多态	68
4.5 项目 4 实现抽象类与接口	75
第 5 章 面向对象编程	79
5.1 项目 1 使用命名空间	79
5.2 项目 2 使用属性	81
5.3 项目 3 使用数组	84
5.4 项目 4 使用枚举	86
5.5 项目 5 实现委托和事件	88
5.6 项目 6 使用索引器	95
5.7 项目 7 使用泛型	97

第 6 章 用户界面与图形图像	102
6.1 项目 1 简单的 Windows 应用程序	102
6.2 项目 2 修改文本框的前景色和背景色	106
6.3 项目 3 设置文本框风格	111
6.4 项目 4 用户界面综合设计	118
6.5 项目 5 Provider 控件使用	130
第 7 章 线程与 I/O 操作	134
7.1 项目 1 创建和终止线程	134
7.2 项目 2 I/O 读写操作	147
第 8 章 ADO.NET 应用基础	157
8.1 数据库基础	157
8.2 什么是 ADO.NET	157
8.3 项目 1 使用 Connection 对象连接数据库	159
8.4 项目 2 使用 Command 操作数据库	163
8.5 项目 3 使用 ADO.NET DataReader	167
8.6 项目 4 使用 DataSet 操作数据	171
第 9 章 窗体应用程序	180
9.1 项目 1 多媒体播放器	180
9.2 项目 2 制作 Web 浏览器	188
第 10 章 ASP.NET 应用基础	196
10.1 安装 SQL Server 2005	196
10.2 项目 制作留言板	199
参考文献	223

第 1 章

C# 语言概述

1.1 C# 简介

1.1.1 C#的产生与发展

C# (读做“C sharp”)是微软公司发布的一种面向对象的、运行于 .NET Framework 之上的高级程序设计语言。其安全、稳定、简单,在继承 C 和 C++ 强大功能的同时去掉了一些它们的复杂特性,并综合了 Visual Basic (简称 VB) 简单的可视化操作和 C++ 运行的高效率,以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的对面向组件编程的支持,成为 .NET 开发的首选语言。

1995 年,微软的竞争对手 SUN 公司首先推出了面向对象的开发语言 Java,提出了跨平台、跨语言的概念,随后 Java 凭借其应用上的巨大优势,迅速成为了企业级应用系统开发的首选工具。为了与其对抗并保持领先地位,微软也推出了自己的基于 Java 语言的编译器 Visual J++。但 SUN 公司认为 Visual J++ 违反了 Java 的许可协议,于是对微软提出了诉讼,这使得微软处于极为被动的局面。出于无奈,微软决定推出其进军互联网的庞大计划 .NET。在一边继续反驳 Sun 公司的控告的同时,微软的程序员们也在技术上进行着一场补救的措施,那就是由 Anders Hejlsberg 领衔开发的 C# 语言,一种非常类似于 Java 的语言。

虽然 C# 看起来与 Java 有着惊人的相似,它包括了诸如单一继承、接口、与 Java 几乎同样的语法和编译成中间代码再运行的过程。但是 C# 与 Java 有着明显的不同,它借鉴了 Delphi 的一个特点,与 COM (组件对象模型) 是直接集成的,而且它是微软公司 .NET Windows 网络框架的主角。经过几年时间的不断开发完善,C# 已成为微软反击 Java 语言的最有力武器。目前,C# 语言最新版本 C# 4.0 已经发布。

回顾 C# 的发展史,C# 1.0 大量模仿了 Java,并保留了 C/C++ 的一些特性,初学者很容易上手;C# 2.0 加入了泛型等重要概念,丰富了自身功能;C# 3.0 加入了大量语法糖,并在没有修改 CLR 的情况下引入了 Linq,非常适合小型程序的快速开发,既减轻了程序员的工作量又提高了代码的可读性;C# 4.0 增加了动态语言的特性,学习了 Python 与 Ruby 语言的优点,开始了静态语言与动态语言的融合。

1.1.2 C#开发环境

Microsoft 公司为 C# 语言推出了两种开发工具:IDE (Visual Studio .NET Integrated Development Environment) 和 SDK (Microsoft .NET Framework Software Development Kit)。SDK 是一个免费的 .NET 平台上的 C# 编译器,其中包含了 C# 编译器、VB.NET 编译器等主要的编译工具和测试工具,但它不能提供 IDE 编译程序的细节。

大多数时候我们采用的 C# 开发环境是 Visual Studio.NET 提供的集成开发环境 IDE。 .NET 集成开发环境 IDE 是提供了多种语言环境的编辑和编译的统一开发平台,它是一个

快速开发企业级 Web 应用程序及高性能桌面应用的优秀工具。目前 .NET 集成开发环境 IDE 的最新版本为 Visual Studio 2010，本书选用的版本为 Visual Studio 2008。在 Visual Studio .NET 集成开发环境下开发 C# 程序，编码易于检查、可以实现动态查找，还可以使用多种工具栏、使用宏语言和对象模型扩展集成开发环境的功能，甚至可以通过自己创建的插件来扩展它的功能。

1. 安装 Visual Studio 2008

将 Visual Studio 2008 安装光盘放入光驱中，自动运行并出现选择安装界面，如图 1-1 所示，单击“安装 Visual Studio 2008”进入安装界面。在 Visual Studio 2008 安装界面中，用户可以根据自身需要选择是否参加微软的帮助改进活动，如图 1-2 所示，单击“下一步”，进入安装程序起始页。



图 1-1 Visual Studio 2008 光盘自动运行界面



图 1-2 Visual Studio 2008 安装界面

在 Visual Studio 2008 安装起始页中，用户可以看到安装程序检测到本机已安装的组件，以及安装程序将要安装的组件。接受许可协议条款，输入安装序列号以及用户名，如图 1-3 所示，单击“下一步”，进入 Visual Studio 2008 安装程序选项页。

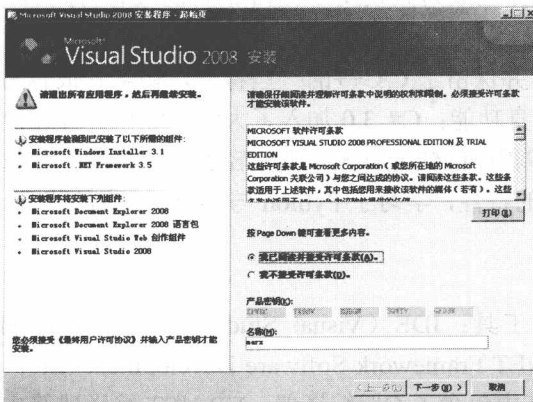


图 1-3 Visual Studio 2008 安装程序起始页

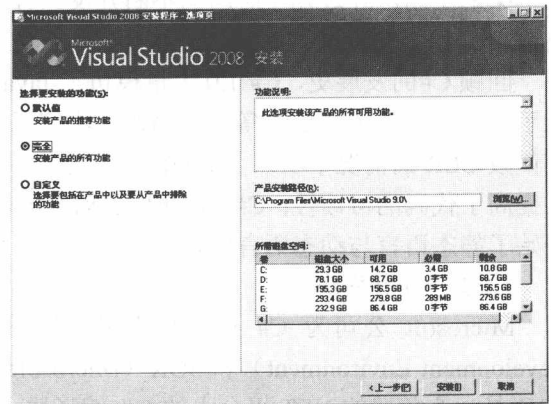


图 1-4 Visual Studio 2008 安装程序选项页

在 Visual Studio 2008 安装程序选项页中，用户可以自行选择安装功能以及安装路径，如图 1-4 所示，单击“安装”按钮，进入 Visual Studio 2008 安装程序安装页，如图 1-5 所示。

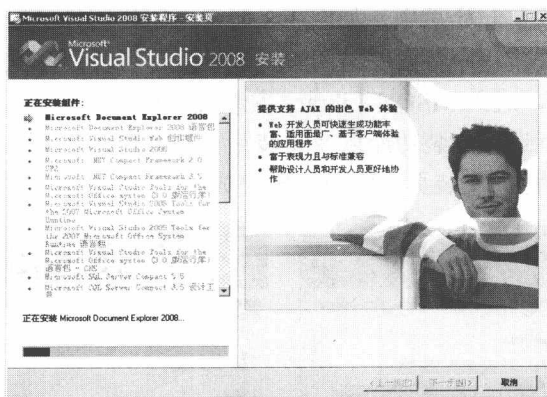


图 1-5 Visual Studio 2008 安装程序安装页

2. 安装 MSDN

安装 Visual Studio 2008 完成后, 安装程序将返回 Visual Studio 2008 安装光盘自动运行界面, 用户可以选择继续安装 MSDN。

MSDN 是 Microsoft Developer Network 的缩写。它是微软公司面向软件开发者的一种信息服务。实际上 MSDN 是一个以 Visual Studio 和 Windows 平台为核心整合的开发虚拟社区, 它包括技术文档、在线电子教程、网络虚拟实验室、微软产品下载 (几乎全部的操作系统、服务器程序、应用程序和开发程序的正式版和测试版, 还包括各种驱动程序开发包和软件开发包)、Blog、BBS、MSDN WebCast 与 CMP 合作的 MSDN 杂志等一系列服务。

此处所安装的 MSDN Library, 只是 MSDN 中的一个产品, 它涵盖了微软全套可开发产品线的技术开发文档和科技文献 (包括部分源代码), 也包括过刊的 MSDN 杂志节选和部分经典书籍的节选章节。Visual Studio 2008 的 MSDN Library 完全安装需要磁盘空间 2.3GB 左右, 如果不安装, 也可以通过在线或者 MSDN 订阅 (需付费) 以脱机方式浏览。在线浏览的链接为 <http://msdn.microsoft.com/zh-cn/default.aspx>。

在 Visual Studio 2008 安装程序页面中, 单击“安装产品文档”进入 MSDN Library 安装界面, 如图 1-6 所示。在 MSDN Library 安装界面中, 用户同样可以根据自身需要选择是否参加微软的帮助改进活动, 如图 1-7 所示, 单击“下一步”, 进入安装程序起始页。



图 1-6 Visual Studio 2008 光盘自动运行界面

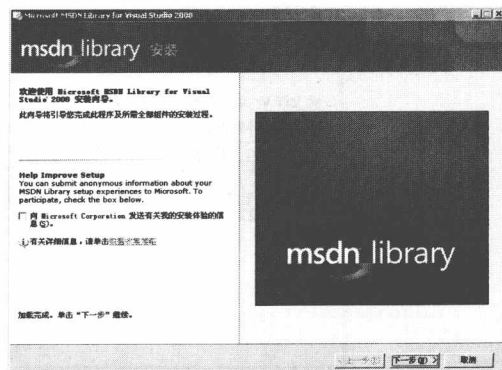


图 1-7 MSDN Library 安装界面

在 MSDN Library 安装起始页中，如图 1-8 所示，用户同样可以看到安装程序检测到本机已安装的组件，以及安装程序将要安装的组件。接受许可协议条款，输入安装序列号以及用户名，单击“下一步”，进入 MSDN Library 安装程序选项页。在 MSDN Library 安装程序选项页中，用户可以自行选择安装功能以及安装路径，如图 1-9 所示，单击“安装”按钮，进入 MSDN Library 安装程序安装页，直至安装完成。

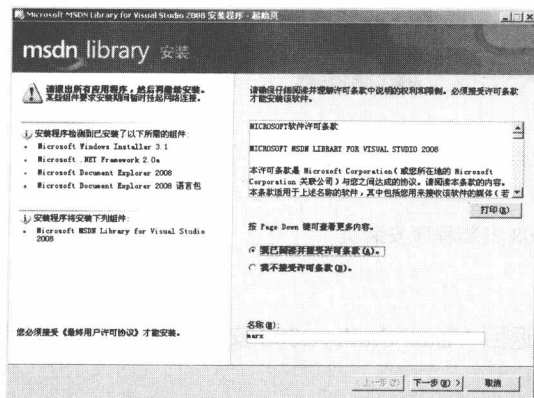


图 1-8 MSDN Library 安装程序起始页

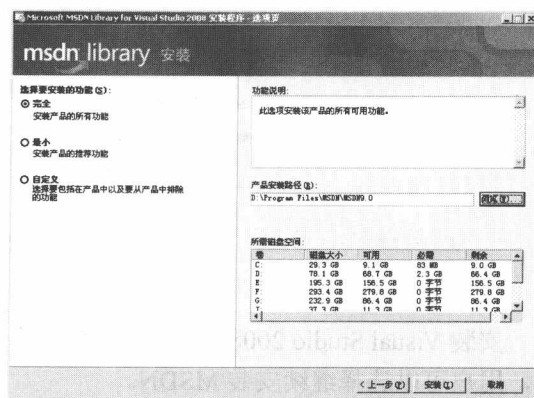


图 1-9 MSDN Library 安装程序选项页

3. 使用 Visual Studio 2008

Visual Studio 2008 安装完成后，首次启动时会出现一个选择默认环境设置对话框，如图 1-10 所示。根据用户的需要，可以选择“Visual Basic 开发设置”、“Visual C#开发设置”、“Visual C++开发设置”、“Web 开发设置”、“常规开发设置”5 项中的一项来设置开发环境。本书主要针对 Visual C#语言进行开发，所以选择“Visual C#开发设置”选项，然后单击启动 Visual Studio 按钮。此时将出现配置进度条，配置过程完成后将显示 Microsoft Visual Studio 起始页，如图 1-11 所示，至此我们就可以使用 Visual Studio 2008 进行 C# 程序的开发了。

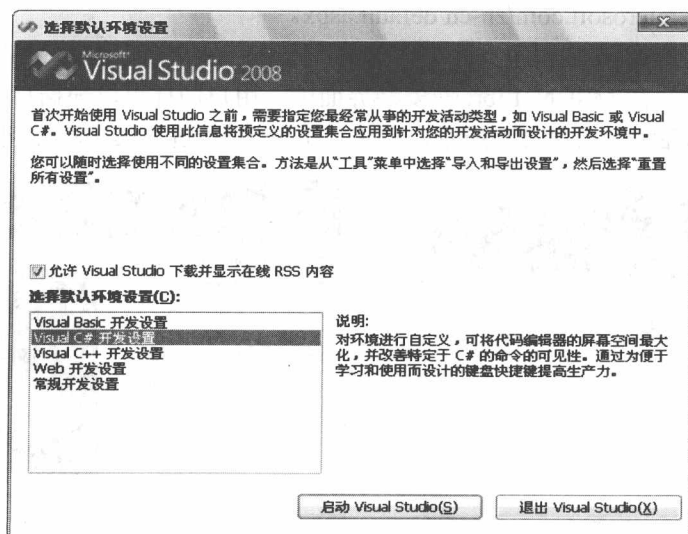


图 1-10 选择默认环境设置界面

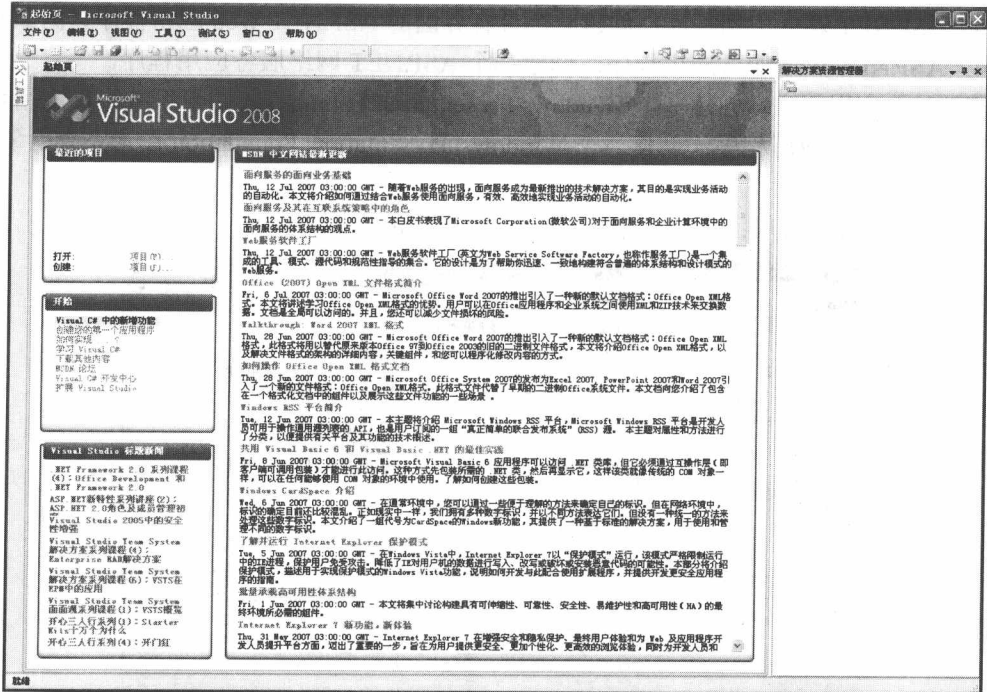


图 1-11 Microsoft Visual Studio 起始页

Visual Studio 2008 起始页包含以下几部分内容。

- (1) 最近的项目：用于显示近期处理过的解决方案列表。另外，这个窗口上还有打开和创建项目的快捷方式。
- (2) 开始：介绍 Visual Studio 2008 下 C# 编程的基础知识，供初学者参考。此功能需要 Internet 连接，如果不能提供网络支持，则可以选择使用本地的微软文档资源管理器。
- (3) Visual Studio 标题新闻：介绍 Visual Studio 最新动向的新闻。此功能需要 Internet 连接，没有网络支持此项功能无法使用。
- (4) MSDN 中文网站最新更新：介绍 Visual Studio 2008 MSDN 中文网站的最新新闻。此项功能同样需要 Internet 连接，没有网络支持此项功能无法使用。

在起始页面中，创建新的项目可以通过选择“文件”→“新建”→“项目”命令来创建，也可以从起始页的最近项目的小标题栏中选择“创建：项目”选项。在 Visual Studio 2008 中，所有的项目都是作为解决方案的一部分来创建的，解决方案中可以包括多个不同类型的项目。解决方案会将用户实例中所用到的全部项目和文件都捆绑在一起。在默认情况下，当创建一个新项目时，当前解决方案会自动关闭，并创建一个新的解决方案保存新创建的项目。

在选择创建一个新的项目时，屏幕会弹出“新建项目”对话框，如图 1-12 所示。其中包括新建项目的类型、使用项目的模板、项目的名称和项目存放的位置。创建一个项目可用的模板很多，每个模板都可以创建一个项目框架，而且还会在项目中添加一组初始源文件。在 Visual Studio 2008 中常用的项目模板主要有如下几个。

- (1) Windows 窗体应用程序：创建一个初始包含单个窗体的 Windows 窗体应用程序。
- (2) 类库：创建一个包含单个 Visual C# 类的项目。

- (3) ASP.NET Web 应用程序：创建一个带有界面的 ASP.NET Web 应用程序，例如网站。
- (4) ASP.NET Web 服务应用程序：创建一个 ASP.NET 网络服务应用程序。
- (5) WPF 应用程序：创建一个 Windows Presentation Foundation 图形系统引擎应用程序。
- (6) WPF 浏览应用程序：创建一个 Windows Presentation Foundation 图形系统引擎浏览器应用程序。
- (7) 控制台应用程序：创建一个 Windows 命令行的应用程序。

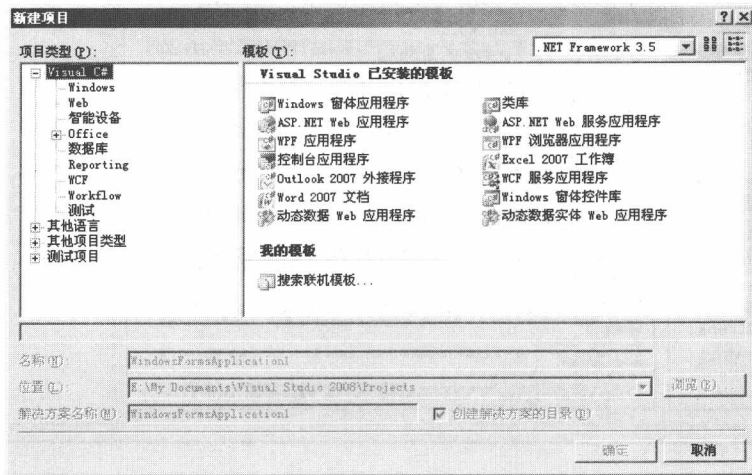


图 1-12 “新建项目”对话框

其他新建项目就不在此处逐一介绍了。创建好所选择的项目后，就可以在代码编辑区编写自己的代码、编译生成目标程序以及运行该程序了。随后我们将演示如何建立 Windows 窗体应用程序、控制台应用程序、Web 站点程序，以及详细介绍开发环境 Visual Studio 2008。

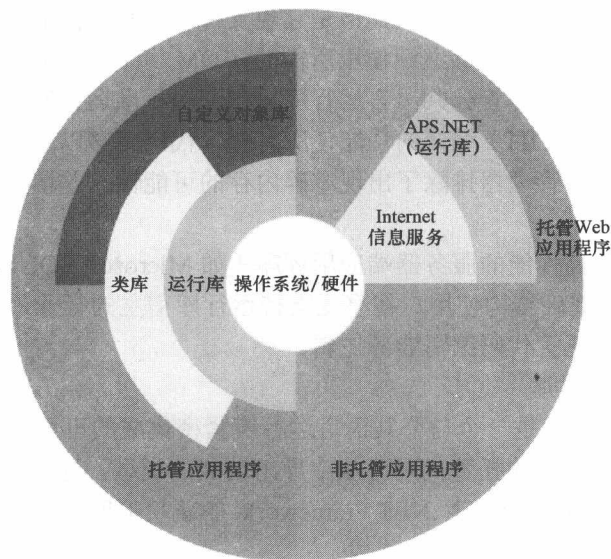
1.2 .NET 应用程序框架

1.2.1 理解 .NET Framework

.NET Framework 是支持生成和运行下一代应用程序及 XML Web Services 的内部 Windows 组件。.NET Framework 旨在实现下述目标。

- (1) 提供一个一致的面向对象的编程环境，而无论对象代码是在本地存储和执行，还是在本地执行但在 Internet 上分布，或者是在远程执行的。
- (2) 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- (3) 提供一个可提高代码（包括由未知的或不完全受信任的第三方创建的代码）执行安全性的代码执行环境。
- (4) 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- (5) 使开发人员的经验在面对类型大不相同的应用程序（如基于 Windows 的应用程序和基于 Web 的应用程序）时保持一致。
- (6) 按照工业标准生成所有通信，以确保基于 .NET Framework 的代码可与任何其他代码集成。

.NET Framework 具有两个主要组件：公共语言运行库和 .NET Framework 类库。公共语言运行库是 .NET Framework 的基础。我们可以将运行库看作一个在执行时管理代码的代理，它提供内存管理、线程管理和远程处理等核心服务，并且还强制实施严格的类型安全以及可提高安全性和可靠性的其他形式的代码准确性。事实上，代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码，而不以运行库为目标的代码称为非托管代码。.NET Framework 的另一个主要组件是类库，它是一个综合性的面向对象的可重用类型集合，我们可以使用它开发多种应用程序，这些应用程序包括传统的命令行或图形用户界面 (GUI) 应用程序，也包括基于 ASP.NET 所提供的最新创新的应用程序 (如 Web 窗体和 XML Web Services)。 .NET Framework 可由非托管组件承载，这些组件将公共语言运行库加载到它们的进程中并启动托管代码的执行，从而创建一个可以同时利用托管和非托管功能的软件环境。 .NET Framework 不但提供若干个运行库宿主，而且还支持第三方运行库宿主的开发。如图 1-13 所示，图中不仅显示公共语言运行库和类库与应用程序之间以及与整个系统之间的关系，而且还显示了托管代码如何在更大的结构内运行。



1.2.2 .NET Framework 的主要组件和功能

1. 公共语言运行库的功能

公共语言运行库管理内存、线程执行、代码执行、代码安全验证、编译以及其他系统服务。这些功能是在公共语言运行库上运行的托管代码所固有的。

至于安全性，取决于包括托管组件的来源（如 Internet、企业网络或本地计算机）在内的一些因素，托管组件被赋予不同程度的信任。这意味着即使用在同一活动应用程序中，托管组件既可能能够执行文件访问操作、注册表访问操作或其他须小心使用的功能，也可能不能够执行这些功能。

运行库强制实施代码访问安全。例如，用户可以相信嵌入到网页中的可执行文件能够在屏幕上播放动画或唱歌，但不能访问他们的个人数据、文件系统或网络。这样，运行库的安

全性功能就使通过 Internet 部署的合法软件能够具有特别丰富的功能。

运行库还通过实现称为通用类型系统（CTS）的严格类型验证和代码验证基础结构来加强代码可靠性。CTS 确保所有托管代码都是可以自我描述的。各种 Microsoft 和第三方语言编译器生成符合 CTS 的托管代码。这意味着托管代码可在严格实施类型保真和类型安全的同时使用其他托管类型和实例。

此外，运行库的托管环境还消除了许多常见的软件问题。例如，运行库自动处理对象布局并管理对对象的引用，在不再使用它们时将它们释放。这种自动内存管理解决了两个最常见的应用程序错误：内存泄漏和无效内存引用。

运行库还提高了开发人员的工作效率。例如，程序员可以用他们选择的开发语言编写应用程序，却仍能充分利用其他开发人员用其他语言编写的运行库、类库和组件。任何选择以运行库为目标的编译器供应商都可以这样做。以 .NET Framework 为目标的语言编译器使得用该语言编写的现有代码可以使用 .NET Framework 的功能，这大大减轻了现有应用程序的迁移过程的工作负担。

尽管运行库是为未来的软件设计的，但是它也支持现在和以前的软件。托管和非托管代码之间的互操作性使开发人员能够继续使用所需的 COM 组件和 DLL。

运行库旨在增强性能。尽管公共语言运行库提供许多标准运行库服务，但是它从不解释托管代码。一种称为实时（JIT）编译的功能使所有托管代码能够以它在其上执行的系统的本机语言运行。同时，内存管理器排除了出现零碎内存的可能性，并增大了内存引用区域以进一步提高性能。

最后，运行库可由高性能的服务器端应用程序 [如 Microsoft SQL Server 和 Internet 信息服务（IIS）] 承载。此基础结构使用户在享受支持运行库宿主的行业最佳企业服务器的优越性能的同时，能够使用托管代码编写业务逻辑。

2. .NET Framework 类库

.NET Framework 类库是一个与公共语言运行库紧密集成的可重用的类型集合。该类库是面向对象的，并提供用户的托管代码可从中导出功能的类型。这不但使 .NET Framework 类型易于使用，而且还减少了学习 .NET Framework 的新功能所需要的时间。此外，第三方组件可与 .NET Framework 中的类无缝集成。

例如，.NET Framework 集合类实现一组可用于开发用户自己的集合类的接口。此集合类将与 .NET Framework 中的类无缝地混合。

正如用户对面向对象的类库所希望的那样，.NET Framework 类型使用户能够完成一系列常见编程任务（包括诸如字符串管理、数据收集、数据库连接以及文件访问等任务）。除这些常见任务之外，类库还包括支持多种专用开发方案的类型。例如，可使用 .NET Framework 开发下列类型的应用程序和服务。

- (1) 控制台应用程序。
- (2) Windows GUI 应用程序（Windows 窗体）。
- (3) Windows Presentation Foundation（WPF）应用程序。
- (4) ASP.NET 应用程序。
- (5) Web 服务。
- (6) Windows 服务。

(7) 使用 Windows Communication Foundation (WCF) 的面向服务的应用程序。

(8) 使用 Windows Workflow Foundation (WF) 的启用工作流程的应用程序。

例如, Windows 窗体类是一组综合性的可重用的类型, 它们大大简化了 Windows GUI 的开发。如果要编写 ASP.NET Web 窗体应用程序, 可使用 Web 窗体类。

1.3 项目显示程序

1.3.1 项目说明

项目 1: 制作简单的控制台应用程序, 在 Windows 控制台中显示“Hello, 天津轻工职业技术学院!”, 如图 1-14 所示。

项目 2: 制作简单的 Windows 窗体应用程序, 在对话框中显示“Hello, 天津轻工职业技术学院!”, 如图 1-15 所示。

项目 3: 制作简单的 ASP.NET 网站, 在网页的文本框中显示“Hello, 天津轻工职业技术学院!”, 如图 1-16 所示。

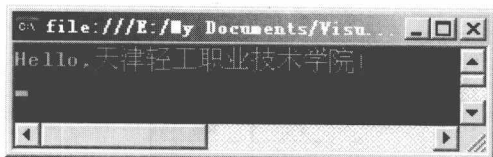


图 1-14 控制台应用程序运行结果



图 1-15 Windows 窗体应用程序运行结果

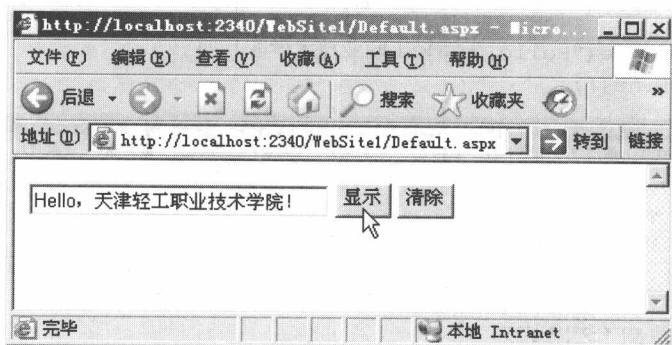


图 1-16 Web 应用程序运行结果

1.3.2 项目分析

项目 1: 在 Windows 控制台中打印输出字符串“Hello, 天津轻工职业技术学院!”, 并且 Windows 控制台不关闭。

项目 2: 在窗体中点击 Hello 按钮, 弹出“Hello, 天津轻工职业技术学院!”对话框, 单击“确定”按钮, 对话框关闭。

项目 3: 在 ASP.NET 网页中, 当单击“显示”按钮时, 文本框中出现“Hello, 天津轻工职业技术学院!”, 当单击清除按钮时, 文本框内的文字消失。

1.3.3 编程实现

1. 项目

打开 Visual Studio 2008, 在起始页面选择“文件”→“新建”→“项目命令”, 如图 1-17 所示。

在弹出的“新建项目”对话框中, 选择 Windows → “控制台应用程序”, 如图 1-18 所示, 输入新建项目的名称以及保存路径后单击确定按钮。

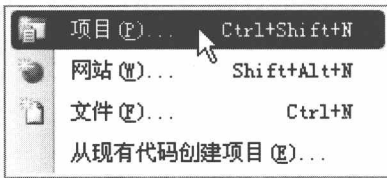


图 1-17 新建项目

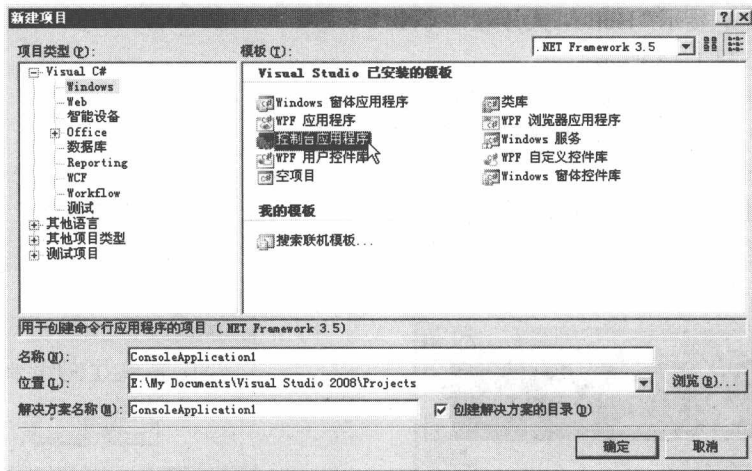


图 1-18 新建控制台应用程序

在页面上 Main 函数中添加如下代码, 如图 1-19 所示。

```
Console.WriteLine("Hello,天津轻工职业技术学院!");
Console.ReadLine();
```

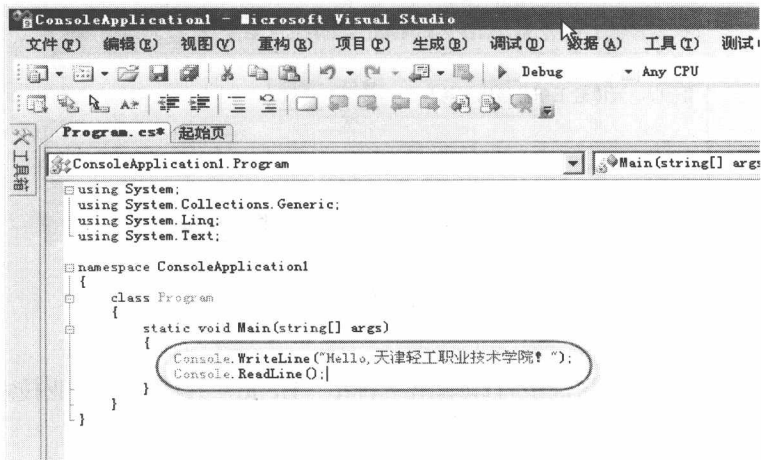


图 1-19 添加代码

按 F5 键启动调试，调试成功后，按 F6 键生成解决方案。运行结果如图 1-20 所示。

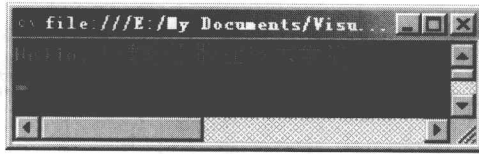


图 1-20 控制台应用程序运行结果

2. 项目 2

打开 Visual Studio 2008，在起始页面选择“文件”→“新建”→“项目”命令，如图 1-21 所示。

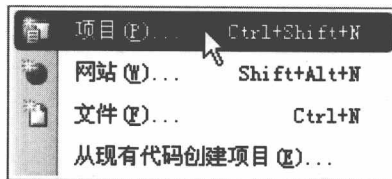


图 1-21 新建项目

在弹出的“新建项目”对话框中，选择 Windows → “Windows 窗体”应用程序，如图 1-22 所示，填好新建项目的名称以及保存路径后单击“确定”按钮。

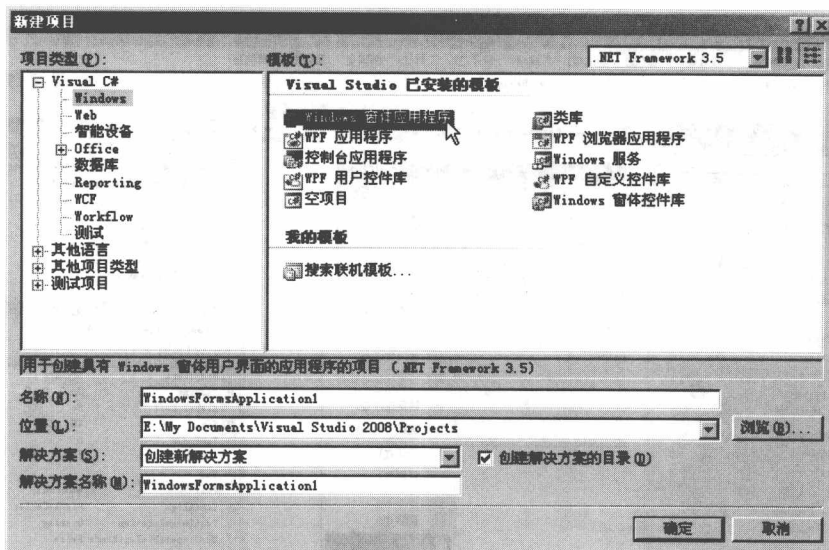


图 1-22 新建 Windows 窗体应用程序

将鼠标移至左边的工具箱上时，Visual Studio 会自动展开工具箱中所有的控件，如图 1-23 所示。工具箱提供了丰富的控件资源供编程人员使用，只需简单的拖拽操作即可将所需要的控件放置到用户界面中。